Experimental results on algebraic attacks on stream ciphers

Frederik Armknecht* Jörg Brandeis† Egor Ilinykh‡
Universität Mannheim

Abstract: Stream ciphers are designed for fast encryption of data of arbitrary length, for example between mobile phones and base stations. Widely used in practice are keystream generators based on linear feedback shift registers (LFSRs). One prominent example is the E_0 generator from the Bluetooth standard for wireless communication.

Algebraic attacks on LFSR-based keystream generators have gained more and more attention in cryptography in the recent years. With these attacks, the secret key is recovered by finding and solving a (possibly large) system of equations. A remarkable fact about algebraic attacks is that the time effort is in the best case only polynomial in the key size.

In this paper, we briefly sketch algebraic attacks and present some experimental results on algebraic attacks on reduced versions of E_0.

Keywords: Keystream generators, E_0, LFSR, algebraic attacks

1 Introduction

Today, electronic communication plays a more and more important role in everybody's life, invoking an increasing demand for confidentiality. Widely used are keystream generators which produce bitstreams $z := z_1, z_2, \ldots$ of arbitrary length in dependence on a (secret) initial value $K \in \{0,1\}^n$. The sender encrypts a stream of plaintext bits $p := p_1, p_2, \ldots$ to a stream of ciphertext bits $c := c_1, c_2, \ldots$ by XOR-ing p and z componentwise, i.e., $c_t := p_t \oplus z_t$. A receiver who uses the same secret key K can produce z by himself and decrypt c_t by $p_t = c_t \oplus z_t$. Following Kerckhoff's principle, it is assumed that an adversary knows the specification of the keystream generator and some of the keystream bits z_t, whereas K is secret to him. Consequently, an attack consists of recovering the secret key K.

2 Algebraic attacks on keystream generators

Due to their efficiency in hardware, keystream generators based on LFSR are widely used in practice. An LFSR is a finite state machine with a linear state transition function which

*armknecht@th.informatik.uni-mannheim.de
†joerg.brandeis@zetvisions.com
‡groebnerbases@gmail.com

outputs one bit of its internal state at each clock. In this paper, we consider the class of combiners with memory, which includes the E_0 keystream generator. A combiner with memory consists of one or several LFSRs and some memory bits. For example, E_0 uses four LFSRs of lengths 25, 31, 33 and 39 and four memory bits. Before keystream generation, the LFSRs and memory bits are initialized to K' and M_0, respectively. At each clock t, some bits $X_t \in \{0,1\}^k$ are extracted from the LFSRs by $L_t(K')$ where L_t is a linear function. The keystream bit z_t is computed by $f(M_t, X_t)$. Finally, the LFSRs are updated according to their state transition functions and the memory bits changed to $M_{t+1} := \delta(M_t, X_t)$.

The security analysis is based on the assumption that an adversary knows the specifications of the keystream generators, particularly the definitions of L_t, f and δ. Further on, it is assumed that he is able to figure out the values of some keystream bits z_t. A straightforward approach, called brute force, is to compute keystreams for all possible values of K' and M_0 and then compare them to the observed data.

Algebraic attacks on memoryless LFSR-based keystream generators have been introduced in [2] and later been extended to combiners with memory in [1]. The first step consists of describing K' by a system of equations.[1] This can be done by using a (preferably low-degree) function F such that the following holds:

$$F(L_t(K'), \ldots, L_{t+r}(K'), z_t, \ldots, z_{t+r}) = 0 \quad \forall t \tag{1}$$

Whereas solving systems of linear equations is feasible using Gaussian elimination, it poses an NP-hard problem in the case of non-linear equations. One of the best studied algorithm so far is based on computing Gröbner bases. Unfortunately, it is still not possible to predict the time effort of the algorithm for a given system of equations, which in the worst case can be exponential in $|K'|$. Unfortunately, the performance of a Gröbner bases approach might heavily depend on the arrangement of the equations what makes a general rigid analysis impossible. Albeit simulations indicate that the time effort decreases with increasing number of equations (see [4]).

In the case of equations as displayed in (1), one can make use of the fact that the degree is less than or equal to the degree of F for all clocks t. Thus, the number of monomials occurring in the system of equations cannot exceed the value $m := \binom{|K'|}{0} + \ldots + \binom{|K'|}{\deg(F)}$. In the case that the number of known keystream bits (and thus the number of equations) is at least m, linearization [5] is the first choice. The idea is to substitute each occurring monomial by a new variable, thereby getting a linear system which is easily solvable by Gaussian elimination. In the best case, i.e., enough keystream bits are known to the adversary, the time effort is in $O\left(|K'|^{\omega \cdot \deg(F)}\right)$ with $\omega \leq 3$.

[1] Since in general, only few memory bits are used, it suffices to recover the value of K'. Once this is done, the value of the memory bits can be easily computed.

Table 1: Experimental results on different attacks on E_0 with reduced key sizes

$\|K'\|$	Brute force			Linearization			Gröbner bases		
	Time	Memory	Data	Time	Memory	Data	Time	Memory	Data
16	11s	3.1 MB	40	$<$ 1m	$\leq$6.04 MB	$\leq 2^{11.3}$	2m 2 s	65.49 MB	350
17	32s	3.1 MB	42	$<$ 1m	$\leq$9.85 MB	$\leq 2^{11.65}$	17m 42 s	180.08 MB	350
18	46s	3.1 MB	44	$<$ 1m	$\leq$15.63 MB	$\leq 2^{11.98}$	39m 23 s	303.37 MB	350
19	1m 28s	3.1 MB	46	$<$ 1m	$\leq$24.19 MB	$\leq 2^{12.3}$	1h 12m	460.27 MB	350
20	3m 44s	3.1 MB	48	$<$ 1m	$\leq$36.61 MB	$\leq 2^{12.6}$	2h 31m	696.63 MB	350
22	18m 13s	3.1 MB	52	52s	$\leq$79.13 MB	$\leq 2^{13.15}$	14 h 14m	2150.33 MB	400
23	20m 33s	3.1 MB	54	1m 9s	$\leq$113.37 MB	$\leq 2^{13.41}$	$>$ 24h	$>$3060.92 MB	500
24	1h 12m	3.1 MB	56	2m 2s	$\leq$159.96 MB	$\leq 2^{13.66}$	-	-	
25	2 h	3.1 MB	58	3m	$\leq$222.55 MB	$\leq 2^{13.9}$	-	-	
26	4h 22m	3.1 MB	60	4m 38s	$\leq$305.64 MB	$\leq 2^{14.13}$	-	-	
27	-	-	-	6m 36s	$\leq$414.74 MB	$\leq 2^{14.35}$	-	-	
28	-	-	-	9m 21s	$\leq$556.57 MB	$\leq 2^{14.56}$	-	-	
29	-	-	-	16m 6s	$\leq$739.21 MB	$\leq 2^{14.76}$	-	-	
30	-	-	-	20m 51s	$\leq$972.36 MB	$\leq 2^{14.96}$	-	-	

3 Experimental results

In this section, we present some experimental results on algebraic attacks on reduced versions of E_0. Instead of using the original LFSRs, shorter LFSRs were used such that $|K'| \ll 128$. Hereby, the function F of degree 4, derived in [1] for the original E_0, was used for all test cases. We tested three different kinds of attacks: brute force, algebraic attacks using the linearization method and using the Gröbner bases method. In the first case, all possible values of K' were tried until the correct one was found. In the second case, we produced keystream bits and created the according equations until the system of equations could be solved by linearization. In the third case, we limited the number of known keystream bits and tried to solve it with an implementation of the Gröbner bases algorithm F4 (see [3]). The results are displayed in table 1. The time effort is given in minutes and seconds, the memory consumption in Megabytes and data is the number of keystream bits assumed to be known.

The results are not directly comparable. The linearization method and the Gröbner bases algorithm have been written in Java, the brute force attack simulated in Magma. Brute force and Gröbner bases have both been simulated on the same computer, a Pentium 4 with 3.5 GHz and 3 Gigabytes RAM. The linearization attack has been simulated on a Pentium 4-M with 1.8 GHz and 512 Megabytes RAM. Since the amount of data and memory has not been recorded during the simulations on linearization, we provide some upper bounds instead. To compute the Gröbner bases, we used the F4 algorithm [3] together with the Becker-Weispfenning-criterion.

Nevertheless, the simulation results give some clues about the attacks behaviour in respect to the key size. As expected, the linearization method is the most successful, outperforming the two other attacks by far. It was possible to derive the secret value K' for $|K'| \leq 30$ within a reasonable amount of time of several minutes. Indeed, instances with key sizes of 34 and higher were intractable due to memory shortness. Further on, the knowledge of several kilobytes of keystream has to be required, a rather unrealistic assumption.

In this regard, Gröbner bases are much more realistic. It was possible to find the value of K' even if only the moderate amount of 350 keystream bits is known. Indeed, we found out that the attack is only feasible if the amount of data is significantly larger than $|K'|$. Even worse, our naive brute force attack implementation proved to be faster and required less data and memory. However, this may be different for larger key sizes.

Another problem are the memory requirements. Since the memory is mainly used to store sparse matrices, this bound may be pushed further by developing appropriate data structures and algorithms. Nevertheless, the experiments indicate that memory is again the bottle neck of this attack.

4 Conclusion

In this paper, we presented simulation results on different types of algebraic attacks and a simple brute force attack on versions of E_0 with reduced key sizes. As expected, a practical attack on the original E_0 is still out of scope on nowadays' computers. Further on, it turned out that memory is the bottle neck of algebraic attacks. Hence, further research could consist in trying to combine the methods from algebraic attacks with guess-and-determine attacks where part of K' is guessed.

References

[1] Frederik Armknecht, Matthias Krause: *Algebraic attacks on Combiners with Memory*, Proceedings of Crypto 2003, LNCS 2729, pp. 162-176, Springer, 2003.

[2] Nicolas Courtois, Willi Meier: *Algebraic attacks on Stream Ciphers with Linear Feedback*, Proceedings of Eurocrypt 2003, LNCS 2656, pp. 345-359, Springer, 2003. An extended version is available at http://www.cryptosystem.net/stream/

[3] Jean-Charles Faugère: *A new efficient algorithm for computing Gröbner bases (F_4)*, Journal of Pure and Applied Algebra 139, 1-3 (1999), 61–68.

[4] Jean-Charles Faugère, Gwénolé Ars: *An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases*, 2003. Available at http://www.inria.fr/rrrt/rr-4739.html.

[5] Adi Shamir, Jacques Patarin, Nicolas Courtois, Alexander Klimov: *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, Proceedings of Eurocrypt '00, Springer LNCS 1807, pp. 392-407.